Exercise 4: Python Functions

Name - Anish Rao

Student Number - 20066423

Group - C

Colab Notebook Link

1. Area of a circle is calculated as follows: area = π x r x r and perimeter = 2 x π x r. Write a function that calculates area_of_circle and perimeter_of_circle.

```
import math
from collections import Counter

def area_of_circle(r):
    return math.pi * r * r

def perimeter_of_circle(r):
    return 2 * math.pi * r

r = float(input("Enter the radius of the circle: "))
print("Area:", area_of_circle(r))
print("Perimeter:", perimeter_of_circle(r))
```

```
Enter the radius of the circle: 7
Area: 153.93804002589985
Perimeter: 43.982297150257104
```

2. Write a function called add_all_nums which takes arbitrary number of arguments and sums all the arguments. Check if all the list items are number types. If not, provide reasonable feedback.

```
def add_all_nums(args):
    for num in args:
        if not isinstance(num, (int, float)):
            return "Error: All arguments must be numbers."
    return sum(args)

nums_input = input("Enter numbers to sum: ").split()
nums = []
for x in nums_input:
  nums.append(float(x))

print("Sum:", add_all_nums(nums))
```

```
Enter numbers to sum: 5 6 7 10 3 4
Sum: 35.0
```

3. Temperature in °C can be converted to °F using this formula: °F = (°C x 9/5) + 32. Write a function which converts °C to °F, convert_celsius_2_fahrenheit.

```
def convert_celsius_2_fahrenheit(c):
    return (c * 9/5) + 32

celsius = float(input("Enter temperature in Celsius: "))
print("Temperature in Fahrenheit:", convert_celsius_2_fahrenheit(celsius))
```

```
Enter temperature in Celsius: 50
Temperature in Fahrenheit: 122.0
```

4. Write a function called check_season, it takes a month parameter and returns the season: Autumn, Winter, Spring, or Summer.

```
def check_season(month):
    if month in [12, 1, 2]:
        return "Winter"
    elif month in [3, 4, 5]:
        return "Spring"
    elif month in [6, 7, 8]:
        return "Summer"
    elif month in [9, 10, 11]:
        return "Autumn"
    else:
        return "Invalid month"

month = int(input("Enter month number (1-12): "))
print("Season:", check_season(month))
```

```
Enter month number (1-12): 6
Season: Summer
```

5. Write a function called calculate_slope which returns the slope of a linear equation.

```
def calculate_slope(x1, y1, x2, y2):
    if x2 - x1 == 0:
        return "vertical line"
    return (y2 - y1) / (x2 - x1)

x1 = float(input("Enter x1: "))
y1 = float(input("Enter y1: "))
x2 = float(input("Enter x2: "))
y2 = float(input("Enter y2: "))
print("Slope:", calculate_slope(x1, y1, x2, y2))
```

```
Enter x1: 4
Enter y1: 8
Enter x2: 2
Enter y2: 6
Slope: 1.0
```

6. Quadratic equation is calculated as follows: ax² + bx + c = 0. Write a function which calculates the solution set of a quadratic equation, solve_quadratic_eqn.

```
def solve_quadratic_eqn(a, b, c):
    if a == 0:
        return "a cannot be 0."
    disc = b**2 - 4*a*c
    if disc > 0:
        root1 = (-b + math.sqrt(disc)) / (2*a)
        root2 = (-b - math.sqrt(disc)) / (2*a)
        return (root1, root2)
    elif disc == 0:
        root = -b / (2*a)
        return (root,)
    else:
        real_part = -b / (2*a)
        imaginary_part = math.sqrt(-disc) / (2*a)
        return (complex(real_part, imaginary_part), complex(real_part, -imaginary_part))

a = float(input("Enter coefficient a: "))
b = float(input("Enter coefficient b: "))
c = float(input("Enter coefficient c: "))
print(f"{a}x² + {b}x + {c} = 0")
print(solve_quadratic_eqn(a, b, c))
```

```
Enter coefficient a: 1
Enter coefficient b: 6
Enter coefficient c: 8
1.0x² + 6.0x + 8.0 = 0
(-2.0, -4.0)
```

7. Declare a function named print_list. It takes a list as a parameter and prints out each element of the list.

```
def print_list(lst):
    for item in lst:
        print(item)

lst_input = input("Enter list items separated by commas: ").split(',')
lst = [item.strip() for item in lst_input]
print("List items:")
print_list(lst)
```

```
Enter list items separated by commas: 5, apple, tree, 9.5
List items:
5
apple
tree
9.5
```

8. Declare a function named reverse_list. It takes an array as a parameter and returns the reverse of the array (use loops).

```
def reverse_list(arr):
    rev = []
    for i in range(len(arr)-1, -1, -1):
        rev.append(arr[i])
    return rev

lst_input = input("Enter list to reverse: ").split(',')
lst = [item.strip() for item in lst_input]
print("Reversed list:", reverse_list(lst))
```

```
Enter list to reverse: apple, tree, egg, banana
Reversed list: ['banana', 'egg', 'tree', 'apple']
```

9. Declare a function named capitalize_list_items. It takes a list as a parameter and returns a capitalized list of items.

```
def capitalize_list_items(lst):
    return [str(item).capitalize() for item in lst]

lst_input = input("Enter list to capitalize: ").split(',')
lst = [item.strip() for item in lst_input]
print("Capitalized list:", capitalize_list_items(lst))
```

```
Enter list to capitalize: elephant, apple, tree
Capitalized list: ['Elephant', 'Apple', 'Tree']
```

10. Declare a function named add_item. It takes a list and an item as parameters. It returns a list with the item added at the end.

```
def add_item(lst, item):
    new_lst = lst.copy()
    new_lst.append(item)
    return new_lst

lst_input = input("Enter initial list: ").split(',')
lst = [item.strip() for item in lst_input]
item_to_add = input("Enter the item to add: ").strip()
print("List after adding item:", add_item(lst, item_to_add))
```

```
Enter initial list: dublin, cork, limerick
Enter the item to add: ireland
List after adding item: ['dublin', 'cork', 'limerick', 'ireland']
```

11. Declare a function named remove_item. It takes a list and an item as parameters. It returns a list with the item removed from it.

```
def remove_item(lst, item):
    new_lst = lst.copy()
    if item in new_lst:
        new_lst.remove(item)
    else:
        print("Item not found in list.")
    return new_lst

lst_input = input("Enter initial list: ").split(',')
lst = [item.strip() for item in lst_input]
item_to_remove = input("Enter the item to remove: ").strip()
print("List after removing item:", remove_item(lst, item_to_remove))
```

```
Enter initial list: dublin, cork, limerick, ireland
Enter the item to remove: ireland
List after removing item: ['dublin', 'cork', 'limerick']
```

12. Declare a function named sum_of_numbers. It takes a number parameter and adds all the numbers in that range.

```
def sum_of_numbers(n):
    return sum(range(1, n+1))

n = int(input("Enter a number to sum from 1 to n: "))
print("Sum of numbers:", sum_of_numbers(n))
```

```
Enter a number to sum from 1 to n: 10
Sum of numbers: 55
```

13. Declare a function named sum_of_odds. It takes a number parameter and adds all the odd numbers in that range.

```
def sum_of_odds(n):
    return sum(i for i in range(1, n+1) if i % 2 != 0)
```

```
n = int(input("Enter a number to sum odd numbers from 1 to n: "))
print("Sum of odd numbers:", sum_of_odds(n))
```

```
Enter a number to sum odd numbers from 1 to n: 6
Sum of odd numbers: 9
```

14. Declare a function named sum_of_even. It takes a number parameter and adds all the even numbers in that range.

```
def sum_of_even(n):
    return sum(i for i in range(1, n+1) if i % 2 == 0)

n = int(input("Enter a number to sum even numbers from 1 to n: "))
print("Sum of even numbers:", sum_of_even(n))
```

```
Enter a number to sum even numbers from 1 to n: 7
Sum of even numbers: 12
```

15. Declare a function named evens_and_odds. It takes a positive integer as a parameter and counts the number of evens and odds in the number.

```
def evens_and_odds(n):
    evens = sum(1 for i in range(n+1) if i % 2 == 0)
    odds = sum(1 for i in range(n+1) if i % 2 != 0)
    print("The number of odds is", odds)
    print("The number of evens is", evens)

n = int(input("Enter a positive integer to count evens and odds: "))
evens_and_odds(n)
```

```
Enter a positive integer to count evens and odds: 10
The number of odds is 5
The number of evens is 6
```

16. Call your function factorial, it takes a whole number as a parameter and returns its factorial.

```
def factorial(n):
    if n < 0:
        return "Factorial of a negative number doesn't exist."
    result = 1
    for i in range(1, n+1):
        result *= i
    return result

n = int(input("Enter a number to compute its factorial: "))
print("Factorial:", factorial(n))
```

```
Enter a number to compute its factorial: 5
Factorial: 120
```

17. Call your function is_empty, it takes a parameter and checks if it is empty or not.

```
def is_empty(param):
    return not bool(param)

param = input("Enter something: ")
print("Is empty?", is_empty(param))
```

```
Enter something: 
Is empty? True
```

18. Write a different function which take lists. They should calculate_mean, calculate_median, calculate_mode, calculate_range, calculate_variance, and calculate_std (standard deviation).

```
def calculate_mean(lst):
    return sum(lst) / len(lst) if lst else 0

def calculate_median(lst):
    n = len(lst)
    if n == 0:
        return None
    sorted_lst = sorted(lst)
    mid = n // 2
    if n % 2 == 1:
        return sorted_lst[mid]
    else:
        return (sorted_lst[mid - 1] + sorted_lst[mid]) / 2

def calculate_mode(lst):
    if not lst:
        return None
    data = Counter(lst)
    mode_val, _ = data.most_common(1)[0]
    return mode_val

lst_input = input("Enter numbers separated by commas for mean, median, mode: ").split(',')
nums = []
for x in lst_input:
    x = x.strip()
    try:
        if '.' in x:
            nums.append(float(x))
        else:
            nums.append(int(x))
    except:
        pass
print("Mean:", calculate_mean(nums))
print("Median:", calculate_median(nums))
print("Mode:", calculate_mode(nums))
```

```
Enter numbers separated by commas for mean, median, mode: 5, 10, 8, 6, 5, 3, 2, 1
Mean: 5.0
Median: 5.0
Mode: 5
```